

学校编码: 10384

密级\_\_\_\_\_

学号: X2006221004

厦 门 大 学

工 程 硕 士 学 位 论 文

面向资源架构(ROA)的  
管理信息系统的设计与实现

Design and Implementation of Management Information  
System Base on Resource-oriented Architecture (ROA)

吴德文

指导教师姓名: 李名世副教授

专 业 名 称: 计算机技术

论文提交日期: 2011 年 月

论文答辩时间: 2011 年 月

学位授予日期

答辩委员会主席:

评阅人:

2011 年 月

## 厦门大学学位论文原创性声明

本人呈交的学位论文是本人在导师指导下,独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果,均在文中以适当方式明确标明,并符合法律规范和《厦门大学研究生学术活动规范(试行)》。

另外,该学位论文为( )课题(组)的研究成果,获得( )课题(组)经费或实验室的资助,在( )实验室完成。(请在以上括号内填写课题或课题组负责人或实验室名称,未有此项声明内容的,可以不作特别声明。)

声明人(签名):

年 月 日

厦门大学博士论文摘要库

## 厦门大学学位论文著作权使用声明

本人同意厦门大学根据《中华人民共和国学位条例暂行实施办法》等规定保留和使用此学位论文，并向主管部门或其指定机构送交学位论文（包括纸质版和电子版），允许学位论文进入厦门大学图书馆及其数据库被查阅、借阅。本人同意厦门大学将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于：

（        ） 1. 经厦门大学保密委员会审查核定的保密学位论文，  
于        年        月        日解密，解密后适用上述授权。

（        ） 2. 不保密，适用上述授权。

（请在以上相应括号内打“√”或填上相应内容。保密学位论文应是已经厦门大学保密委员会审定过的学位论文，未经厦门大学保密委员会审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权。）

声明人（签名）：

年        月        日

厦门大学博士论文摘要库

## 摘 要

采用 SOA 实现独立信息系统间的无缝对接是当今信息工程领域的前沿技术和研究热点。SOA 也是下一代互联网在应用层面的主要特征，并且是云计算 SaaS 的核心技术。掌握这一关键技术将意味着进入当今信息工程的前沿技术领域，具有广阔的发展空间。

SOA 是在 Web Service 的基础上发展起来的架构模型，而 Web Service 主要使用 SOAP 协议进行通讯。然而，SOAP 仍属于复杂的、重量级的协议。随着当前 Web 2.0 的兴起，表述性状态转移（REST）逐步成为一个流行的架构风格。REST 是一种轻量级的 Web Service 架构风格，其性能、效率和易用性都优于基于 SOAP 协议的 Web Service 架构。REST 的核心概念是资源，基于 REST 的 Web Service 构建了面向资源的架构（ROA）。随着 REST 逐渐成为 Web Service 的主流技术，ROA 也渐渐地成为 SOA 的最有前途的轻量型替代方案。

本文的研究内容旨在为一个基于 ROA 架构的项目构建基础框架。通过实现这个基础框架，完整地探索在 Java EE 平台上实现最新的 RESTful Web Service 的技术路线和细节，从而为进入这一前沿技术领域开辟一条道路，奠定发展的基础。

项目的应用原型是厦门大学信息学院实验教学中心的管理信息系统。系统构建在 Java EE 平台上并采用基于 REST 的 ROA 架构进行设计，其先进技术特色体现在如下几个方面：

1. 采用 JPA 技术进行对象关系映射，屏蔽数据库产品之间的差异；
2. 使用 Apache Commons Digester 解析 XML 数据文件；
3. Web 应用通过 RESTful Web Service 存取数据，而不是直接操作数据库；
4. 规范化 RESTful Web Service 的数据格式，降低操作不同数据的差异性；
5. 使用 EJB 3.0 进行 AOP 设计；
6. 使用基于单点登录技术的校园统一身份认证平台进行身份验证。

本文的研究在上述几个方面实现了当今信息工程的前沿技术，其技术成果可以更加广泛地应用于校内外独立信息系统的互联互通和信息资源整合。

**关键词：**REST；ROA；RESEasy；EJB；JPA；

厦门大学博硕士论文摘要库

## Abstract

Nowadays, it is cutting-edge technology and hot research area of information engineering that connect each individual information system seamlessly by using the service-oriented architecture (SOA). SOA is the main character of the next generation Internet applications, and it is the core technology of the SaaS (Software as a Service) in cloud computing. Mastering the SOA technology will enter the cutting-edge of today's information engineering and developing in a more wide space.

SOA is an architecture base on web service, and communicate with the web service via the Simple Object Access Protocol (SOAP). Basically, SOAP is still a complicated and heavy-weight protocol. As the Web 2.0 rising up, Representational State Transfer (REST) has become a popular style web service. The core of REST is resource. RESTful web service is lightweight, and its performance, efficiency and ease of use are superior to the one based on SOAP. RESTful web services build resource-oriented architecture (ROA). While the RESTful web service becomes more popular, ROA has gradually become the most promising and lightweight alternative solution of SOA.

The target of this paper is to build a framework that provide a set RESTful API of data operation for a project based on ROA. As the sample of the API, this paper need implement a simple module also. Through the implementation of this basic framework, exploring the latest technology and full details of RESTful web service on Java EE, so as to open up a road to this cutting edge, lay the foundation for the development.

The prototype of the project is a MIS of Experimental Teaching Center of Information School of Xiamen University. The final goal of this project is manage all kind of information in the work. The MIS is built on Java EE, and is designed as ROA architecture. It has the following advanced features:

1. ORM by JPA, hiding the difference between DB software;



2. Parse XML data file by Apache Commons Digester;
3. Web App R/W data through RESTful Web Service instead of DB;
4. Standardize the data format of RESTful Web Service;
5. AOP with EJB 3.0
6. Authenticate base single sign-on technology.

These features are implemented by leading-edge technology of information engineering nowadays. The achievement can be more widely used to integrate information system of school.

**Key Words:** REST; ROA; RESTEasy; JPA; EJB;

# 目 录

摘要.....	i
英文摘要 .....	ii
目录.....	iv
英文目录 .....	vii
<b>第一章 绪论.....</b>	<b>1</b>
<b>1.1 研究背景及意义.....</b>	<b>1</b>
1.1.1 研究内容.....	3
<b>1.2 论文组织结构.....</b>	<b>4</b>
<b>第二章 ROA 架构的相关理论技术.....</b>	<b>5</b>
<b>2.1 Web Service .....</b>	<b>5</b>
2.1.1 Web Service 概述 .....	5
2.1.2 Web Service 基本规范 .....	6
2.1.3 Web Service 的使用方式 .....	7
<b>2.2 SOA 实现技术.....</b>	<b>8</b>
2.2.1 SUN 的 JBI.....	9
2.2.2 IBM 的 SCA.....	9
<b>2.3 REST 技术规范及实现 .....</b>	<b>10</b>
2.3.1 REST 简介.....	10
2.3.2 REST 技术规范.....	12
2.3.3 REST 实现框架.....	12
<b>2.4 数据持久化技术.....</b>	<b>14</b>
2.4.1 序列化.....	14
2.4.2 JDBC .....	15
2.4.3 关系对象映射（Object Relational Mapping, ORM） .....	15
2.4.4 EJB 2.X CMP(Container-Managed Persistence).....	16
2.4.5 Java 数据对象（Java Data Object, JDO） .....	16
2.4.6 Hibernate .....	16
2.4.7 JPA(Java Persistence API) .....	16
<b>第三章 ROA 架构设计与技术工具.....</b>	<b>18</b>
<b>3.1 系统需求分析.....</b>	<b>18</b>
3.1.1 业务功能模块.....	18
3.1.2 公共支撑模块.....	19

3.1.3	系统角色.....	20
3.1.4	数据存取模式.....	20
3.1.5	系统界面.....	21
<b>3.2</b>	<b>系统架构设计.....</b>	<b>21</b>
3.2.1	与 Web Service 交互的应用客户端 .....	21
3.2.2	RESTful 的 iDATA 数据访问接口 .....	22
3.2.3	接口与实现分离的数据访问层.....	23
3.2.4	多样化的数据存储层.....	24
<b>3.3</b>	<b>数据库设计 .....</b>	<b>24</b>
<b>3.4</b>	<b>开发工具选用.....</b>	<b>28</b>
3.4.1	编程语言.....	28
3.4.2	Java EE 服务器-JBoss .....	29
3.4.3	REST 框架.....	29
3.4.4	数据持久化技术-JPA .....	29
3.4.5	数据库-MySQL.....	29
3.4.6	Eclipse .....	30
<b>第四章</b>	<b>ROA 架构具体实现.....</b>	<b>31</b>
<b>4.1</b>	<b>基于 JPA 的数据持久化.....</b>	<b>31</b>
4.1.1	配置数据库.....	31
4.1.2	配置 JPA 项目 .....	32
4.1.3	实体 Bean .....	33
4.1.4	实体关联.....	35
4.1.5	实体 CRUD 操作 .....	38
<b>4.2</b>	<b>使用 Commons Digester 装载 XML 数据文件.....</b>	<b>40</b>
4.2.1	定义 XML 规则文件 .....	40
4.2.2	扩展 Digester 的 MenuDigester .....	41
<b>4.3</b>	<b>基于泛型设计的 DAO 抽象类.....</b>	<b>43</b>
<b>4.4</b>	<b>AOP 与 EJB 3.0 会话 Bean.....</b>	<b>45</b>
4.4.1	@Local 和 @Remote .....	45
4.4.2	@Stateless.....	45
4.4.3	@EJB 注入会话 Bean .....	46
<b>4.5</b>	<b>RESTful 的 iDATA API.....</b>	<b>47</b>
4.5.1	安装 RESTEasy.....	47
4.5.2	定义资源.....	48
4.5.3	定义基 URI.....	50
4.5.4	定义 CRUD 操作 .....	51
4.5.5	@PathParam、@FormParam 和 @QueryParam.....	52
4.5.6	定义响应的 MIME 类型.....	53
<b>4.6</b>	<b>REST 的客户端代码 .....</b>	<b>54</b>
4.6.1	基于 JSAPI 的 Ajax .....	54
4.6.2	HTML 页面 .....	56

<b>4.7 基于单点登录技术的身份验证 .....</b>	<b>57</b>
4.7.1 校园一卡通单点登录系统说明.....	57
4.7.2 抽象化的单点登录提供者.....	58
4.7.3 校园统一身份认证平台的代理.....	59
4.7.4 单点登录截图.....	60
<b>第五章 总结及展望 .....</b>	<b>62</b>
<b>参考文献 .....</b>	<b>63</b>
<b>致谢.....</b>	<b>64</b>

厦门大学博硕士论文摘要库

# CONTENT

<b>Abstract .....</b>	<b>ii</b>
<b>CONTENT .....</b>	<b>vii</b>
<b>Chapter 1 Introduction .....</b>	<b>1</b>
<b>1.1 Background of the project.....</b>	<b>1</b>
<b>1.2 Content and structure of paper .....</b>	<b>4</b>
<b>Chapter 2 Theory &amp; Tech. About ROA.....</b>	<b>5</b>
<b>2.1 Web Service .....</b>	<b>5</b>
2.1.1 Overview of web service .....	5
2.1.2 Protocols of web service .....	6
2.1.3 Usage of web service .....	7
<b>2.2 Implementation of SOA.....</b>	<b>8</b>
2.2.1 JBI of SUN .....	9
2.2.2 SCA of IBM .....	9
<b>2.3 REST Introduction .....</b>	<b>10</b>
2.3.1 Overview of REST .....	10
2.3.2 Technics of REST .....	12
2.3.3 REST frameworks .....	12
<b>2.4 Data Persistence Technology .....</b>	<b>14</b>
2.4.1 Serialize .....	14
2.4.2 JDBC .....	15
2.4.3 Object Relational Mapping, ORM .....	15
2.4.4 EJB 2.X CMP(Container-Managed Persistence).....	16
2.4.5 Java Data Object, JDO .....	16
2.4.6 Hibernate .....	16
2.4.7 Java Persistence API, JPA .....	16
<b>Chapter 3 ROA System Design &amp; Dev. Tools .....</b>	<b>18</b>
<b>3.1 Requirement Analysis .....</b>	<b>18</b>
3.1.1 Core business module .....	18
3.1.2 Public module .....	19
3.1.3 System roles .....	20
3.1.4 Data access mode .....	20
3.1.5 UI .....	21
<b>3.2 System Architecture .....</b>	<b>21</b>
3.2.1 Client level.....	21

3.2.2	RESTful data access portal(iData API)	22
3.2.3	Data access level	23
3.2.4	Data store level	24
<b>3.3</b>	<b>Design of Database</b>	<b>24</b>
<b>3.4</b>	<b>Tools</b>	<b>28</b>
3.4.1	Java language	28
3.4.2	JBoss server	29
3.4.3	RESEasy framework	29
3.4.4	JPA	29
3.4.5	MySQL	29
3.4.6	Eclipse	30
<b>Chapter 4</b>	<b>Features of Project</b>	<b>31</b>
<b>4.1</b>	<b>JPA Based Data Persistence</b>	<b>31</b>
4.1.1	Config Database	31
4.1.2	Config JPA project facet	32
4.1.3	Entity bean	33
4.1.4	Entity relationship	35
4.1.5	Entity CRUD operation	38
<b>4.2</b>	<b>Load XML Data File by Commons Digester</b>	<b>40</b>
4.2.1	Define XML rule file	40
4.2.2	MenuDigester extended Digester	41
<b>4.3</b>	<b>Generic Abstract DAO</b>	<b>43</b>
<b>4.4</b>	<b>AOP and EJB 3.0 Session Bean</b>	<b>45</b>
4.4.1	@Local & @Remote	45
4.4.2	@Stateless	45
4.4.3	@EJB injection	46
<b>4.5</b>	<b>RESTful iDATA API</b>	<b>47</b>
4.5.1	Install RESEasy	47
4.5.2	Define resource	48
4.5.3	Define base URI	50
4.5.4	Define CRUD operation	51
4.5.5	@PathParam, @FormParam and @QueryParam	52
4.5.6	Define MIME type of response	53
<b>4.6</b>	<b>Client Code of RESTful Service</b>	<b>54</b>
4.6.1	Ajax using REST JSAPI	54
4.6.2	HTML Page	56
<b>4.7</b>	<b>Authenticate by Single Sign-on</b>	<b>57</b>
4.7.1	Usage of Compus Single Sign-on Platform	57
4.7.2	Abstract CheckIn Provider	58
4.7.3	Proxy of Compus Single Sign-on Platform	59
4.7.4	Screenshot	60

**Chapter 5 Conclusion ..... 62**

**References ..... 63**

**Thanks ..... 64**

厦门大学博硕士论文摘要库



Degree papers are in the "[Xiamen University Electronic Theses and Dissertations Database](#)". Full texts are available in the following ways:

1. If your library is a CALIS member libraries, please log on <http://etd.calis.edu.cn/> and submit requests online, or consult the interlibrary loan department in your library.
2. For users of non-CALIS member libraries, please mail to [etd@xmu.edu.cn](mailto:etd@xmu.edu.cn) for delivery details.

厦门大学博硕士论文摘要库